

REMARKS

In view of the following remarks, Applicant respectfully requests withdrawal of the rejections of the claims that are rejected.

Priority

The office action states that Applicant has received the benefit of priority of an earlier filing date to application 60/158,164, filed October 7, 1999. Applicant believes this is in error.

Applicant has timely claimed priority to Application Serial No. 09/304,133 (now issued Patent No. 6,654,741). However, Applicant has not claimed priority to application 60/158,164.

Information Disclosure Statement

Applicant acknowledges that the information disclosure statements filed on July 22, 2003 have been considered as to the merits.

Double Patenting

Claims 1 - 7, 9, and 12 have been rejected under the judicially created doctrine of obviousness-type double patenting as being purportedly unpatentable over claims 1, 3, 5, 9, 37, and 39 of U.S. Patent No. 6,654,741 B1. A terminal disclaimer in compliance with 37 CFR 1.321(c) is filed herewith. Applicant requests that the rejections of claims 1 - 7, 9, and 12 under obviousness-type double patenting be withdrawn.

BEST AVAILABLE COPY

1 **Claim Objections**

2 Claim 12 is objected to because it is incorrectly numbered. As shown
3 above in the Amendments to Claims, claim 12 has been renumbered to claim 11.
4 Applicant requests that the objection to claim 12 be withdrawn.
5

6 **§102(e) Rejections**

7 Claims 1, and 3-12 stand rejected under 35 U.S.C. §102(e) as being
8 anticipated by U.S. Patent No. 6,119,078 to Kobayakawa et al. (hereinafter
9 "Kobayakawa"). Applicant traverses these rejections and, for the following
10 reasons, respectfully urges the Office to reconsider the rejections.

11 Claim 1 recites a method of mapping a URL string comprising searching
12 for a particular pattern in an input URL string, *the pattern being defined in a*
13 *manner that permits the search to be satisfied while allowing variability among*
14 *constituent parts of the input URL string*, and replacing the input URL string with
15 an output URL string if the pattern is found in the input URL string. Kobayakawa
16 neither discloses nor suggests any such method. In fact, for reasons that will be
17 noted below, Kobayakawa teaches directly away from the recited subject matter.

18 In support of its rejection of claim 1, the Office argues that Kobayakawa
19 discloses: (1) a particular pattern in an input URL string (citing to col. 5, lines 26 –
20 53, col. 6, lines 1 – 34, col. 11, lines 1 – 67, col. 12, lines 1 – 67); (2) a pattern
21 defined in a manner that permits the search to be satisfied (citing to col. 5, lines 25
22 – 53, col. 6, lines 1 – 9, col. 8, lines 41 – 57, col. 11, lines 1 – 66); and (3) while
23 allowing variability among constituent parts of the input URL string (col. 5, lines
24 25 – 53, col. 6, lines 1 – 9, col. 8, lines 41 57, col. 11, lines 1 – 67, col. 10, lines 1
25

1 - 11). After a careful and thorough study of Kobayakawa, Applicant respectfully
2 disagrees with the Office.

3 Kobayakawa discloses a method in which a Web page in a first language
4 can be translated into a second language by interpreting the URL used to request
5 the Web page. A character string is located that is similar to at least a portion of
6 the URL. A translating environment is linked to the located character string and
7 automatically translates the Web page transmitted from a server. The translated
8 Web page may be displayed in both the first and second languages.

9 According to Kobayakawa's translating method, a database for URL
10 character strings is created with a format, such as that illustrated in Table 1
11 appearing directly below.

TABLE 1

URL character strings	
Translating environments	
Web.ibm.com/	Translating environment A (Internet dictionary)
Web.yahoo.com/	Translating environment A (Internet dictionary)
Web.yahoo.com/Arts/	Translating environment B (art dictionary)
Web.yahoo.com/Arts/Recreation/Sports	Translating environment C (sports dictionary)

20 The database includes URL character strings for Web pages accessed
21 relatively frequently and translating environments associated with the URL
22 character strings. The translating environments that correspond to respective URL
23 character strings are regarded as appropriate environments, based on the results
24 obtained by the past performances of the translation process.
25

1 As set forth in Kobayakawa, in Table 1, translating environment A (an
2 Internet dictionary) is recorded as an appropriate environment for URL names
3 "Web.ibm.com/" and "Web.yahoo.com/". Translating environment B (an art
4 dictionary) is recorded as an appropriate environment for "Web.yahoo.com/Arts".
5 Translating environment C (a sports dictionary) is recorded as an appropriate
6 environment for "Web.yahoo.com/Arts/Recreation/ Sports/".

7 When a user inputs a URL character string to a Web browser screen, the
8 Web browser (or the proxy of the Web browser) downloads a file from a Web
9 page designated in the URL character string. A computer process then interprets
10 the input URL character string. An interpretation of the URL character string
11 refers to a process employed to locate a URL character string that most nearly
12 resembles (i.e., is most analogous to) the input URL character string. More
13 specifically, a URL character string is regarded as a combination of partial URL
14 character strings. A URL character string having the combination of partial URL
15 character strings that most nearly resembles the input URL character string is
16 searched for in the database. When, for example, the input URL is
17 "Web.ibm.com/", which is in the database, the corresponding translating
18 environment A (an Internet dictionary) is automatically selected. If the input URL
19 string is "Web.yahoo.com/Arts/Architecture/", which is not registered in the
20 database, translating environment B (an art dictionary), which corresponds to
21 "Web.yahoo.com/Arts/", whose partial URL character strings most closely
22 resemble the input URL string, is automatically selected.

23 The process just described does not in any way employ a searching step in
24 which a search for a particular pattern is conducted, where the particular pattern is
25 defined in a manner which permits *variability* among its constituent parts. Rather,

1 Kobayakawa's method performs a character by character search using the input
2 URL character string as the pattern that it is looking for. There is no variability
3 whatsoever in the pattern that Kobayakawa is looking for. This is necessarily so
4 because Kobayakawa's method is looking for the "URL character string that most
5 nearly resembles the input URL character string." In order to find the URL
6 character string that most nearly resembles the input URL character string,
7 Kobayakawa must perform an exacting, character-by-character comparison where
8 *no variability* is permitted in the input URL string. Were this not the case,
9 Kobayakawa could conceivably be inoperative because it would not, in some
10 cases, select the appropriate translating environment. Accordingly, for at least this
11 reason, claim 1 is not anticipated by Kobayakawa.

12 In addition, the Office argues that Kobayakawa discloses replacing the
13 input URL string with an output string if the pattern is found in the input URL
14 string (citing to Fig. 3, elements 110-135, col. 9, lines 30 - 39, col. 10, lines 12 -
15 26, col. 11, lines 10 - 67, col. 12, lines 1 - 5). Applicant has carefully studied
16 Kobayakawa and, in particular, the passages cited by the Office to support this
17 specific rejection. Applicant can find no disclosure or suggestion of any process
18 that replaces the input URL string with an output string if the pattern is found in
19 the input URL string.

20 For example, column 11, lines 10-58 are set forth directly below:

21
22 If, at decision block S40, $m \neq 0$ (i.e., if a similar URL character string
23 has already been temporarily stored), program control moves to block S50.
24 At block S50, the URL character $S(i)$ that is currently being used for
25 comparison is compared with the URL character string $S(m)$ already stored
to determine whether the URL character string more nearly resembles the
currently input URL character string S . This determination is performed by
comparing the URL character strings $S(i)$ and $S(m)$ to decide which has a

1 greater number of sequential partial URL character strings. For
2 "Web.yahoo.com/" and "Web.yahoo.com/Arts/", since the former consists
3 of one partial URL character string and the latter consists of two, the latter
4 is superior.

5 The phrase "a greater number of sequential partial URL character
6 strings" means that even when more partial URL character strings
7 correspond to each other, unless the arrangement of the partial URL
8 character strings do not match, the similarity is low as the result of the URL
9 comparison. Assume that a currently input URL character string is
10 "Web.yahoo.com/ Recreation/Sports/" and two URL character strings,
11 "Web.yahoo.com/Sports/" and "Web.yahoo.com/Recreation/" are registered
12 in the URL database 133. Although the latter have two sequential partial
13 URL character strings that match those of the input URL character string,
14 the former has only one partial URL character string that matches that of
15 the input URL character string (although "Sports" is included in the input
16 URL character string, it does not count because the arrangements do not
17 correspond to each other). Therefore, the latter URL character string is
18 superior.

19 When, at decision block S50, it is ascertained that the newly
20 compared URL character string S(i) is better, at block S60, i is substituted
21 into m in order to change the most analogous URL character string. Then,
22 program control goes to block S70. If, at decision block S50, the URL
23 character string S(m) previously stored is better, program control skips
24 block S60, to maintain the current value for m, and goes to block S70.

25 At block S70, i is incremented by one to compare the next record in
the URL database 133, and program control thereafter returns to block S20.
If, at block S20, an uncompared record remains in the URL database 133,
the process at blocks S30 through S70 is repeated. When, at block S20, it is
ascertained that the value held by current index i has exceeded the count for
the last record in the URL database 133, program control branches to "Yes"
and goes to block S80.

The foregoing passage describes a procedure shown in Fig. 6 of
Kobayakawa. The procedure is a routine performed by the URL comparison
section 134 for selecting a translating environment. An index, 'i' is used to step
through a number of registered URLs, S(i). When one of the URLs S(i) is found
that resembles an input URL, S, another index 'm' takes on the value of 'i' to

1 temporarily store the URL character string used for comparison. There is no
2 disclosure or suggestion throughout the foregoing passage that the input URL, S,
3 is replaced with any of the URLs, S(i). The procedure of Fig. 6 is merely used to
4 find a translating environment that corresponds to the input URL.

5 This can be clearly seen in column 11, lines 59-67 through column 12, lines
6 1-5, which are set forth directly below:

7
8
9 At block S80, a check is performed to determine whether $m=0$.
10 When $m=0$, it means that, at the loop S20 through S70, a URL character
11 string that resembles the currently input URL character string S has not
12 been found in the URL database 133. In this case, a default translating
13 environment is output to the translation engine 120 (block S90), and the
14 routine for selecting a translating environment is terminated.

15 If $m \neq 0$, the translating environment E(m) for the URL character
16 string regarded as most analogous at the loop S20 through S70 is output to
17 the translation engine 120 (block S100). The routine for selecting a
18 translating environment is thereafter terminated. The translation engine 120
19 employs the selected translating environment E(m) to translate a
20 downloaded file.

21 This passage above describes how a particular translating environment is
22 selected based upon the input URL character string being searched. If a particular
23 URL string is not found exactly in the database, then a translating environment
24 associated with the closest URL character string is selected. Nothing in this
25 passage discloses or suggests a method that replaces the input URL string with an
output string if the pattern is found in the input URL string. In fact, the claimed
"replacing" step is unnecessary to achieve Kobayakawa's intended purpose, which
is to select a translating environment.

Column 9, lines 31-67 are set forth directly below:

FIG. 3 is a schematic diagram illustrating an arrangement of the translating environment selector 130. The translating environment selector 130 is subdivided into a URL input section 131, a translating environment output section 132, a URL database 133, a URL comparison section 134, and a basic operating section 135. The URL database 133 is a table which contains URL character strings for Web pages accessed relatively frequently, and a corresponding translating environment for each character string.

Referring now to FIGS. 4 and 5, the URL database 133 displayed on a Web browser screen is illustrated (FIG. 5 is a continuation of the screen in FIG. 4). The URL character strings for Web servers/Web pages accessed frequently are registered in the database 133, in correlation with their appropriate translating environments. For example, "Web.ibm.com/" and "Web.yahoo.com/" are registered in conjunction with their appropriate translating environment "Internet, general", while "cnn.com/" is registered in conjunction with its appropriate environment "politics". In the illustrated URL database 133, the respective character strings are described irregularly (i.e., described only in the order in which they were registered, not in any other order such as alphabetical, partial URL character string arrangement, or string count).

One characteristic point concerning the relationship between a URL character string and its corresponding translating environment is that when the URL character strings describe the same Web server name but the designated files differ, in many cases the appropriate translating environment also differs. As is shown in FIGS. 4 and 5, for example, "Internet, general" is registered as a corresponding environment for "Web.yahoo.com/". However, "politics", "sports" and "art" are registered as appropriate environments for files in the same Web server, "Web.yahoo.com/ Government/", "Web.yahoo.com/Recreation/Sports/", and "Web.yahoo.com/Arts/" respectively.

Nothing in this passage discloses or suggests a method that replaces the input URL string with an output string if the pattern is found in the input URL string.

Column 10, lines 12-26 are set forth directly below:

A user of the automatic translation system 100 may register in advance URL character strings in the URL database 133. In addition, a user

1 of the system 100 may also register the URL character strings via the basic
operating section 135, which will be described later.

2 The URL input section 131 receives, via the translation proxy 110, a
3 URL character string that is input on the Web browser screen, and transmits
4 it to the URL comparison section 134. The URL comparison section 134
5 performs a comparison process for the input URL character string on the
6 URL database 133, and finds an appropriate translating environment. The
URL comparison section 134 reports the comparison result to the
translation engine 120 through the translating environment output section
132.

7 Again, nothing in this passage cited by the Office discloses or suggests a
8 method that replaces the input URL string with an output string if the pattern is
9 found in the input URL string. As discussed above, the purpose of Kobayakawa is
10 to select a translating environment. Therefore, replacing the input URL string
11 with an output string if the pattern is found in the input URL string is irrelevant to
12 the intended purpose of Kobayakawa.

13 Accordingly, for at least the foregoing reasons, claim 1 is neither disclosed
14 nor suggested by Kobayakawa and is allowable.

15 Claims 2-5 depend either directly or indirectly from claim 1 and are
16 allowable as depending from an allowable base claim. These claims are also
17 allowable for their own recited features which, in combination with those recited
18 in claim 1, are neither shown nor suggested by the references of record, either
19 singly or in combination with one another.

20 Claim 6 recites a Uniform Resource Locator (URL) mapping engine
21 comprising an Application Programming Interface (API) that exposes a plurality
22 of methods that are associated with managing rules that govern mapping
23 capabilities of the URL mapping engine. Kobayakawa neither discloses nor
24 suggests any such mapping engine. In fact, for reasons that will be noted below,
25 Kobayakawa teaches directly away from the recited subject matter.

1 In support of its rejection of claim 6, the Office argues that Kobayakawa
2 discloses: (1) a Uniform Resource Locator (URL) mapping engine (citing figure 2,
3 element 120) comprising (2) an Application Programming Interface (API) (citing
4 col. 8, lines 40 – 47) that (3) exposes a plurality of methods that are associated
5 with managing rules that govern mapping capabilities of the URL mapping engine
6 (citing col. 8, lines 58 – 67; col. 9, lines 1 – 39; col. 10, lines 27 – 48; col. 11, lines
7 59 – 67, col. 12, lines 1 – 6 and 53 – 67, and col. 13, lines 1- 4). After a careful
8 and thorough study of Kobayakawa, Applicant respectfully disagrees with the
9 Office.

10 Preliminarily, the Applicant would like to point out that, based on an
11 electronic search performed on Kobayakawa through the Office's web site,
12 Applicant has found that Kobayakawa makes no mention of the terms "application
13 program interface," "application programming interface," or "API." In addition,
14 Kobayakawa mentions "application program" only once at col. 8, line 33.
15 Furthermore, Kobayakawa mentions "interface" only twice (at col. 1, line 31 and
16 col. 8, line 43), and these two occurrences are in reference to a graphical user Web
17 browsing interface. As a result, Kobayakawa clearly does not literally disclose an
18 API.

19 Despite this, the Office asserts that Kobayakawa discloses an API at
20 column 8, lines 40 – 47, which are set forth directly below:

21 The Web browser 150 is a computer program that acts as an Internet
22 tour guide. The Web browser 150 provides a user with a user interface for
23 inputting a URL. Also, the Web browser 150 interprets the contents of an
24 HTML file included in the Web page and displays the result of the
25 interpretation on the display 22. The Web browser 150 in this embodiment
has a proxy function.

1
2 The above passage mentions a user interface and a web browser that
3 interprets the contents of a Web page. According to "The Computer Glossary," a
4 user interface is "the combination of menus, screen design, keyboard commands,
5 command language and help screens, which create the way a user interacts with a
6 computer." See *The Computer Glossary*, Alan Freedman, Ninth Edition, The
7 Computer Language Company Inc., © 2001, p. 420 (copy attached). By contrast,
8 an API is "a language and message format used by an application program to
9 communicate with another program that provides services for it." See *id.*, p. 11
10 (copy attached). Thus, an API assists two application programs in
11 communicating. The cited passage mentions a user communicating with a single
12 application program. Clearly, the cited passage neither discloses nor suggests an
13 API, but rather a well-known user interface for interacting with a Web browser.

14 Even if it were assumed, *arguendo*, that Kobayakawa discloses an API,
15 Kobayakawa does not disclose or suggest an API that exposes a plurality of
16 methods that are associated with managing rules that govern mapping capabilities
17 of the URL mapping engine. For example, column 8, lines 58 – 67 and column 9,
18 lines 1 – 39 are shown below:

19
20 Upon a request from the Web browser 150, the translation proxy 110
21 downloads, via the Internet 70, a file from a Web page designated by the
22 URL, and transmits text in the downloaded file to the translation engine
23 120. The translation proxy 110 also receives the translation of text from the
24 translation engine 120 and transmits it to the Web browser 150. The
25 "proxy" is an agent for a host that cannot perform a direct outside access,
and is the generic term for a relay function, such as the reception of
communication data from the Internet.

The translation engine 120 facilitates performing a translation of text
in a foreign language to text in a native language. By employing a

1 translating environment, the translation engine 120 translates the original
2 text that has been received from the translation proxy 110, and transmits the
3 translation of the text to the translation proxy 110. The translation engine
4 120 in this embodiment has a plurality of translating environments, such as
5 translating environment A, translating environment B and translating
6 environment C. A "translating environment" is a dictionary database and
7 grammatical algorithms (e.g., setups for clauses, setups for auxiliary verb
8 meanings, and sentence stylistic designations) that are used for the
9 translation process. As is generally known, there exists correlation between
10 a genre, a field and an application of original text and a translating
11 environment. The present invention provides for various genre a respective
12 special dictionary database, such as an Internet dictionary, a business
13 dictionary, a politics dictionary, an entertainment dictionary, an art
14 dictionary, a sports dictionary, and the like.

9 The translating environment selector 130 facilitates selection of a
10 translating environment to be used by the translation engine 120. The
11 design of the translating environment selector 130 in this embodiment is
12 based on the fact that a URL consists of a combination of character string
13 segments (i.e., "partial URL character strings") that respectively describe a
14 protocol name, a server name and a file name. The translating environment
15 selector 130 automatically selects an appropriate translating environment in
16 accordance with a URL character string input by the Web browser 150.

13 FIG. 3 is a schematic diagram illustrating an arrangement of the
14 translating environment selector 130. The translating environment selector
15 130 is subdivided into a URL input section 131, a translating environment
16 output section 132, a URL database 133, a URL comparison section 134,
17 and a basic operating section 135. The URL database 133 is a table which
18 contains URL character strings for Web pages accessed relatively
19 frequently, and a corresponding translating environment for each character
20 string.

19 The foregoing passage simply discusses Kobayakawa's translating engine
20 120 that selects a translating environment and translates text of a foreign language
21 to text in a native language. There is no mention of an API that exposes a plurality
22 of methods that are associated with managing rules that govern mapping
23 capabilities of the URL mapping engine.

24 Column 10, lines 27 - 48 of Kobayakawa are set forth directly below:
25

1 The URL comparison section 134 searches the URL database 133
2 for a URL character string that is similar to (i.e., most nearly resembles) an
3 input URL character string, and outputs to the translation engine 120 a
4 translating environment that corresponds to the obtained URL character
5 string. FIG. 6 is a flowchart for a routine performed by the URL
6 comparison section 134 when selecting a translating environment. In this
7 routine, S denotes a URL character string input by the Web browser, i
8 denotes an index on the URL database 133, and m denotes an index for the
9 URL character string obtained during the search process that S most nearly
10 resembles. S(i), for example, represents a URL character string that is
11 registered in the i-th record in the URL database 133. When the final result
12 obtained by the database search is $m=M$, it means that the appropriate
13 translating environment is the one that corresponds to the M-th URL
14 character string S(M) in the URL database 133. E(m) represents a
15 translating environment for a URL character string that is registered in the
16 m-th record in the URL database 133.

17 Again, the foregoing passage simply discusses Kobayakawa's translating
18 engine 120 that selects a translating environment and translates text of a foreign
19 language to text in a native language based on a URL that resembles an input
20 URL. There is no mention of an API that exposes a plurality of methods that are
21 associated with managing rules that govern mapping capabilities of the URL
22 mapping engine.

23 The cited column 11, lines 59 – 67 and column 12, lines 1 – 6 are discussed
24 above. These cited sections fail to disclose or suggest an API that exposes a
25 plurality of methods that are associated with managing rules that govern mapping
capabilities of the URL mapping engine.

Cited column 12, lines 53 – 67, and col. 13, lines 1- 4 are set forth directly
below:

A URL input portion 208 is provided immediately under the tool bar
206. A user can enter a URL character string in the URL input portion 208
to designate a desired Web page to be accessed. In FIG. 7, URL name

1 "Web.yahoo.com/" is input. The input URL character string is transmitted
2 to the translation proxy 110. The translation proxy 110, which takes the
3 place of the Web browser 150, downloads a designated Web page and
4 temporarily transmits the original text in the Web page to the translation
5 engine 120. The input URL character string is transmitted to the translating
6 environment selector 130 via the translation proxy 110. The translating
7 environment selector 130 interprets the received URL character string and
8 selects for it an appropriate translating environment, which is reported to
9 the translation engine 120. The translation engine 120 employs the selected
10 translating environment to translate original text and transmits the original
11 text and the translated text to the Web browser 150 via the translation proxy
12 110. This processing sequence may be performed in the background of a
13 desktop.

14 Again, the foregoing passage simply discusses Kobayakawa's translating
15 engine 120 that selects a translating environment and translates text of a foreign
16 language to text in a native language. There is no mention of an API that exposes
17 a plurality of methods that are associated with managing rules that govern
18 mapping capabilities of the URL mapping engine.

19 For at least the foregoing reasons, Kobayakawa fails to disclose or suggest
20 all of the limitations of claim 6. Claim 6 is therefore allowable.

21 Claim 7 recites a method of mapping a URL string. In accordance with the
22 method, an input URL string is received and mapped to an output expression
23 having a tagged expression. The tagged expression is used to provide an output
24 URL string. The passages cited by the Office in making out this rejection simply
25 do not disclose or suggest receiving an input URL string, mapping the input URL
string to an output expression having a tagged expression, and using the tagged
expression to provide an output URL string. Accordingly, for at least this reason,
claim 7 is allowable.

1 Claim 8 depends from claim 7. Claim 8 is therefore allowable for at least
2 the same reasons as claim 7. In addition, claim 8 includes other novel and
3 nonobvious limitations that are not taught or suggested by Kobayakawa. For these
4 reasons, claim 8 is allowable.

5 Claim 9 is directed at a computer-readable medium having computer-
6 executable instructions for performing acts including receiving an input Uniform
7 Resource Locator (URL) string, evaluating the input URL string against a plurality
8 of rules to identify a rule specifying a text pattern corresponding to the URL
9 string, each rule having an output expression associated therewith, at least one
10 rules specifying a text pattern correspond to more than one combination of text
11 characters, and producing an output URL string using an output expression
12 associated with the identified rule.

13 In support of its rejection of claim 9, the Office argues that Kobayakawa
14 discloses: at least one rules specifying a text pattern correspond to more than one
15 combination of text characters (citing col. 4, lines 62 – 67, col. 5, lines 1 – 53). In
16 the present application, a rule defines a special character for specifying variability
17 in an input expression (See Application, p. 11, lines 12 – 14). An exemplary set of
18 rules is shown in Fig. 3. Kobayakawa neither discloses nor suggests a plurality of
19 rules, wherein at least one of the rules specifies a text pattern corresponding to
20 more than one combination of text characters, as recited in claim 9.

21 For example, cited col. 4, lines 62 – 67 through col. 5, lines 1 – 53 are set
22 forth directly below:

23 As is known by those skilled in the art of Web client/server
24 communications, the functionality of a hypertext document (referred to
25 hereinafter as a "Web page") comes from its ability to link text, images, and
other objects within a document to text, images, and objects located
elsewhere on the Internet. A Web page can be comprised of text, images

1 and a variety of objects, each of which are surrounded by various tags
2 which control format attributes and identify different portions of the
document (for example: <tag_name>text</tag_name>). Web pages are
typically written and stored in ASCII text format using a text editor.

3 HyperText Markup Language uses so-called "markup tags," denoted
4 by the <> symbols, with the actual tag between the brackets. Most markup
tags have a beginning tag (<tag>) and an ending tag (</tag>). For example,
5 to make a line of text appear as a heading the following tags enclose the
text: <H3></H3>. (<H3>This text appears as a heading.</H3>). To make a
6 line of text appear as a larger heading the following tags enclose the text:
<H2></H2>. (<H2>This text appears as a larger heading.</H2>). To make
7 a word or line of text appear in bold the text is enclosed by the tags:
. (Bold text). In addition there are numerous link tags in
8 HTML to enable the viewer of a Web page to jump to another place in the
same page, to jump to a specific place in another Web page, or to create and
9 jump to a remote link (via a new URL) to another Web server. The HTML
language is described in the HTML Reference Manual, published by Sandia
10 National Laboratories, and available on the Internet at
11 "http://Web.sandia.gov/sci-compute/html.ref.html", which is incorporated
herein by reference, in its entirety. It is to be understood that the terms "tag"
12 and "markup tag" can be used interchangeably.

13 A "translating environment" includes a dictionary database and
grammatical algorithms (e.g., setups for clauses, setups for auxiliary verb
14 meanings, and sentence stylistic designation), which are used for translation
processing. The present invention is intended mainly for the automatic
15 translation of Web pages, and is based on the fact that a Web page URL
consists of a combination of descriptive character string segments (i.e.,
16 "partial URL character strings): a protocol name, a server name and a file
name.

17 The present invention can be implemented via a computer process,
18 which is performed by interlocking with a series of browsing operations for
receiving a Web page from a Web server and for displaying it on a browser
19 screen. The computer process includes a translation engine, used for
translating foreign language text into native language text (or vice versa),
20 and a plurality of translating environments, such as translating environment
A (Internet dictionary), translating environment B (art dictionary), and
21 translating environment C (sports dictionary).

22 According to an automatic translating method of the present
invention, a database for URL character strings is created with a format,
23 such as that illustrated in Table 1. The illustrated database includes URL
character strings for Web pages accessed relatively frequently and
24 translating environments associated therewith. The translating
environments that correspond to respective URL character strings are
25

1 regarded as appropriate environments, based on the results obtained by the
2 past performances of the translation process.

3 The foregoing passage in no way discloses or suggests a rule specifying a
4 text pattern corresponding to more than one combination of text characters. The
5 majority of the passage (i.e., col. 4, lines 62 – 67 through col. 5, lines 1 – 25)
6 simply discusses conventional Web pages and HTML. The remainder of the
7 passage, col. 5, lines 26 – 53, discuss a translating environment for translating web
8 pages from one language to another based on an input URL string. Nowhere in
9 the passage is there any mention of a rule specifying text pattern corresponding to
10 more than one combination of text characters. Indeed, nowhere does Kobayakawa
11 disclose a rule specifying text pattern corresponding to more than one combination
12 of text characters

13 For the foregoing reasons alone, claim 9 is not anticipated by Kobayakawa.
14 In addition, claim 9 includes other limitations that are neither disclosed nor
15 suggested by Kobayakawa. Therefore, claim 9 is allowable. Claim 10 depends
16 from claim 9 and is therefore allowable for at least the same reasons as claim 9.

17 Referring to claim 12 (now claim 11), claim 12 is set forth directly below:

18
19 A computer-readable medium having computer-executable
20 instructions for performing acts comprising:
21 defining a plurality of rules, wherein each rule specifies:
22 a text pattern;
23 a rule ID;
24 a rule action type; and
25 a corresponding output expression;
wherein at least some of the text patterns correspond to more than
one combination of text characters;
evaluating the rules against a URL string to identify a rule specifying
a text pattern corresponding to the URL string; and

1 replacing the URL string with an output string specified by the
2 output expression of the identified rule.

3 In its rejection of claim 12, the Office does not specifically set forth the
4 reasons for this rejection. The Office simply states that "the limitations of this
5 claim has been noted in the rejection above. It is therefore rejected as set forth
6 above." Apparently, the Office is basing the rejection of claim 12 on arguments
7 similar to those in support of rejection of other claims.

8 Although there may be some overlap between the features of claim 12 and
9 other claims, claim 12 is not identical to any other claim. Rather, the Applicant
10 has claimed a number of features as an attempt to conform to the Office's
11 preferred practice of submitting claims having a range of breadth. The Applicant
12 has submitted extra fees for the inclusion and examination of claim 12. In
13 response to this effort, it is the Office's responsibility to fully examine claim 12,
14 and to give full consideration to each of the limitations of claim 12 with grounds
15 of rejection for each limitation of claim 12. See, e.g., MPEP 706.02(j).

16 Kobayakawa clearly does not disclose or suggest all the limitations of claim
17 12. Because Kobayakawa does not employ rules, Kobayakawa has no need for,
18 and does not disclose, defining a plurality of rules, wherein each rule specifies a
19 text pattern, a rule ID, a rule action type, and a corresponding output expression,
20 wherein at least some of the text patterns correspond to more than one
21 combination of text characters. Furthermore, Kobayakawa does not disclose
22
23
24
25

1 evaluating rules against a URL string to identify a rule specifying a text pattern
2 corresponding to the URL string.

3 In addition, for at least the reasons given above with respect to claim 1,
4 Kobayakawa does not replace a URL string with an output string specified by the
5 output expression of the identified rule. As discussed above, the purpose of
6 Kobayakawa is to select a translating environment, and not to replace an input
7 URL string with an output expression.

8
9 For at least the foregoing reasons, claim 12 (now claim 11) is allowable
10 over Kobayakawa.

11 **§103(a) Rejections**

12 Claim 2 stands rejected under 35 U.S.C. §103(a) over Kobayakawa in view
13 of U.S. Patent No. 6,094,649 to Bowen. Applicant traverses this rejection for at
14 least the following reasons.

15 Because claim 2 depends from claim 1, claim 2 is believed to be allowable
16 for at least the same reasons as claim 1. Claim 2 includes other novel and
17 nonobvious elements. Specifically, claim 2 recites the method of claim 1, wherein
18 the particular pattern comprises a regular expression. A regular expression (as
19 described in Applicant's specification on page 10, lines 9-15) comprises a
20 character string in which literal characters indicate text that must exist identically
21 in an input URL string.

22 In support of the 103(a) rejection, the Office cites column 5, lines 55 – 67,
23 which are reproduced below:
24
25

1 As used here, a "keyword" search is a pattern-matching search which
2 tries to locate instances of digital data using a key word or phrase. Many
3 conventional web search engines support keyword searches. Keywords may
4 contain wildcards. For instance, if the question mark is used as a wildcard
5 capable of matching any single character and the asterisk is used as a
6 wildcard capable of matching any zero or more characters, then the
7 keyword "b?t*" would match the words "bat", "bet", "bit", "bot", "but",
8 "battle", "bitten", and "butane", among others. In some cases keywords may
9 also contain regular expressions, such as the regular expressions used in the
10 familiar lexical analysis program lex or the familiar text editors emacs and
11 vi. A keyword may contain smaller keywords connected by operators such
12 as AND and OR.

13 Bowen refers to "regular expression" once in the entire patent at column 5,
14 line 66, shown above. Bowen does not clearly define what is meant by "regular
15 expression", except by stating that examples are regular expressions used by
16 lexical analysis program lex or text editor emacs and vi. Therefore, it is unclear
17 whether Bowen's "regular expression" is the same as the regular expression
18 recited in claim 2.

19 Even if it were assumed, *arguendo*, that Bowen's regular expression
20 corresponds to a regular expression as used in claim 2, there is no motivation to
21 combine Bowen with Kobayakawa. Bowen and Kobayakawa serve very different
22 purposes. Kobayakawa translates a web page based on a URL input by the user.
23 By contrast, Bowen associates keywords with resource locators. In addition, the
24 user of Bowen inputs a keyword and not a URL. By contrast, the user of
25 Kobayakawa already knows and types in a URL string. (See Kobayakawa at col.
6, line 10, "When the user inputs a URL character string..."). Therefore, there is
no motivation to combine Bowen and Kobayakawa.

In addition, in claim 2, the pattern comprises a regular expression, but it is
the URL string that is input. By contrast, in Bowen, the user inputs the keyword

1 that can contain Bowen's regular expression and the URL string is determined. As
2 such, Bowen in no way suggests the elements of claim 2. Therefore, Kobayakawa
3 and Bowen, either separately or in combination, fail to disclose or suggest all of
4 the elements of claim 2.

5 For at least the reasons given above, claim 2 is believed to be allowable.
6 Applicant respectfully requests withdrawal of the rejection of claim 2.

7
8 New Claims

9 Claims 13 - 20 have been added and is allowable over the cited references.
10 New claims 13 - 20 add no new matter and are clearly allowable over the art of
11 record.

12
13 Conclusion

14 All of the claims are in condition for allowance. Accordingly, Applicant
15 respectfully requests that a Notice of Allowability be issued forthwith. If the
16 Office's next anticipated action is to be anything other than issuance of a Notice of
17 Allowability, Applicant respectfully requests a telephone call for the purpose of
18 scheduling an interview.

19
20 Respectfully Submitted,

21
22 Dated: 8/26/04

By: 

23 Damon A. Rieth
24 Reg. No. 52,167
25 (303) 539-0265 ext. 237

user**user**

Any individual who interacts with the computer at an application level. Programmers, operators and other technical personnel are not considered users when working in a professional capacity on the computer.

user defined

Any format, layout, structure or language that is developed by the user.

user friendly

A system that is easy to learn and easy to use. This term has been so abused that many vendors are reluctant to use it.

user group

An organization of users of a particular hardware or software product. Members share experiences and ideas to improve their understanding and use of a particular product. User groups are often responsible for influencing vendors to change or enhance their products.

user interface

The combination of menus, screen design, keyboard commands, command language and help screens, which create the way a user interacts with a computer. Mice, touch screens and other input hardware is also included. A well-designed user interface is vital to the success of a software package. In time, interactive video, voice recognition and natural language understanding will be included.

USRT

(Universal Synchronous Receiver Transmitter) An electronic circuit that transmits and receives data on the serial port. It converts bytes into serial bits for transmission, and vice versa, and generates the necessary signals for synchronous transmission.

utility program (utilities)

A program that supports using the computer. Utility programs, or "utilities," provide file management capabilities, such as sorting, copying, comparing, listing and searching, as well as diagnostic and measurement routines that check the health and performance of the system.

UTP

See *twisted pair*.

UTF

(Universal Transformation Format) A method for converting 16-bit Unicode characters into 7-bit or 8-bit characters. UTF-7 converts to 7-bit ASCII for transmission over 7-bit mail systems, while UTF-8 converts Unicode to 8-bit bytes. See *Unicode* and *7-bit ASCII*.

UUCP

(UNIX to UNIX CoPy) A UNIX utility that copies a file from one computer to another. It is commonly used as a mail transfer. Unlike TCP/IP, which is a routable communications protocol, UUCP provides a point-to-point transmission where a user at one UNIX computer dials up and establishes a session with another UNIX computer.

UUencode (UUcoding)

A method for encoding binary files for transmission via Internet e-mail, which was originally designed for ASCII text. UUencode and UUdecode were the first methods. Today, MIME is widely used. See *BinHex* and *MIME*.

UUNET

(UUNET Technologies, Inc., Fairfax, VA, www.uunet.net) Founded in 1987, UUNET was a commercial Internet service provider. Originally offering e-mail and news, it is now a full-service organization providing dial-up and leased line accounts as well as archive space for files and web pages. UUNET stands for UNIX to UNIX Network. In 1996, UUNET was acquired by MFS Communications, which itself was acquired by WorldCom, Inc., in that same year. WorldCom was acquired by MCI.

Apple key**ANSI terminal**

A display terminal that follows commands in the ANSI standard terminal language. Uses escape sequences to control the cursor, clear the screen and set colors, for example. Communications programs often support the ANSI terminal.

anti-aliasing

In computer graphics, a category of techniques that is used to smooth the jagged appearance of diagonal lines. For example, the pixels that surround the edges of the line are filled in with varying shades of gray or color in order to blend the sharp edge into the background. See *dithering*.

antivirus

A program that detects and removes a virus.

AOL

(America OnLine) The country's largest online service. AOL provides Internet access, conferencing, news, e-mail, education and support forums. Specialized software for Windows and Mac provide navigation through the system.

Apache

(A "patchy" server) A widely-used public domain, UNIX-based Web server from the Apache Group (www.apache.org). It is based on NCSA's HTTPd server. The name came from a body of existing code and many "patch files."

API

(Application Program Interface) A language and message format used by an application program to communicate with another program that provides services for it. APIs are usually implemented by writing function calls. Examples of APIs are the calls made by an application program to such programs as an operating system, messaging system or database management system (DBMS). See *interface*.

APL

(A Programming Language) A high-level, scientific language noted for its brevity and matrix generation capabilities. Developed by Kenneth Iverson in the mid 1960s, it is often used to develop mathematical models. More popular in Europe, APL uses unique character symbols and requires a special font to display and print them.

APM

(Advanced Power Management) An API from Intel and Microsoft for battery-powered computers that lets programs communicate power requirements to slow down and speed up components.

APPC

(Advanced Program-to-Program Communications) A high-level communications protocol from IBM that allows a program to interact with another program. It supports client/server and distributed computing by providing a common programming interface across all IBM platforms for communications over a variety of transport protocols. It provides commands for managing a session, sending and receiving data and transaction security and integrity (two-phase commit).

Apple

(Apple Computer, Inc., Cupertino, CA, www.apple.com) A manufacturer of personal computers and the industry's most fabled story. Founded in a garage by Steve Wozniak and Steve Jobs in 1976 and guided by Mike Markkula, Apple blazed the trails for the personal computer industry.

From its Apple II series to the Macintosh to today's new PowerMacs, Apple has always provided a unique alternative to personal computing. The Macintosh's graphical user interface, which was introduced in 1984, set the standard for ease of use that is unmatched.

Apple II

The personal computer family from Apple that pioneered the microcomputer revolution and was widely used in schools and home. It used the 8-bit 6502 microprocessor running at 1MHz, an 8-bit bus and ran Apple's DOS or ProDOS operating system.

Apple key

The original name of the Command key.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.